

# Assignment 1: ROS 2 Publisher and Subscriber (C++)

Robotics Applications Course

## Objective

In this assignment, you will implement two ROS 2 nodes in **C++17 (ROS 2 Humble)**:

- A **publisher node** that publishes an integer counter at a fixed rate.
- A **subscriber node** that subscribes to the topic, processes the integer to square it, and logs the result.

This assignment will help you understand the ROS 2 publisher–subscriber communication model.

## Tasks

1. Create a new ROS 2 package for this lab using the `ament_cmake` build system.
2. Inside the package, implement two C++ nodes:
  - A **publisher node** that generates and publishes an integer counter at a fixed frequency (e.g., once per second).
  - A **subscriber node** that receives the published integer, processes it by squaring the value, and prints the result to the console.
3. Build the package within your ROS 2 workspace so that both nodes can be executed.
4. Launch the publisher and subscriber nodes in separate terminals so they can communicate with each other.
5. Verify that the communication is working correctly by checking the list of available topics and observing the data being transmitted on the relevant topic.

## Submission Requirements

Each student must submit:

1. A public GitHub repository named: `ros2-lab1-pubsub-{rollnumber}` containing:
  - Source code (`publisher.cpp`, `subscriber.cpp`)
  - `CMakeLists.txt` and `package.xml`

- A `README.md` with:
  - ROS 2 version (**Humble**, C++17)
  - Build instructions
  - Run instructions
- 2. A **PDF report** containing:
  - Short description of the objective.
  - The GitHub repository link.
  - Step-by-step build and run instructions.
  - Screenshots of publisher-subscriber communication.
  - A brief conclusion (what you learned).

## Evaluation Criteria

- **Correctness (40%)**: Publisher and subscriber nodes work as expected.
- **Implementation (30%)**: Proper use of ROS 2 APIs and clean C++ code.
- **Documentation (20%)**: Clear README and report.
- **Demonstration (10%)**: Ability to explain the workflow in report .